

LAPORAN RESMI
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK
ENKAPSULASI



Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
TA 2022/2023

A. TUGAS PENDAHULUAN

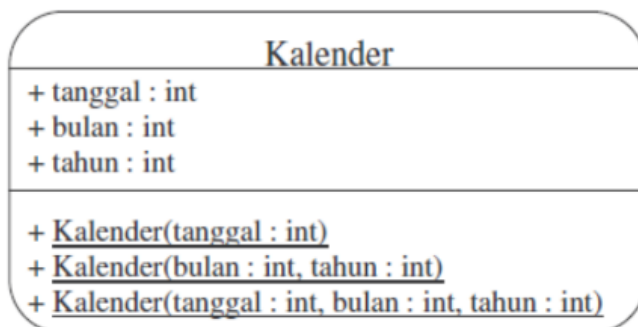
1. Apakah yang dimaksud dengan enkapsulasi?
Enkapsulasi adalah suatu cara untuk menyembunyikan implementasi detail dari suatu class. Enkapsulasi memiliki dua hal mendasar yaitu
 - a. Information hiding
 - b. Menyediakan suatu method untuk mengakses data.
2. Apakah yang dimaksud dengan constructor?
Constructor adalah suatu method yang pertama kali dijalankan pada saat pembuatan suatu obyek. Constructor mempunyai ciri – ciri yaitu
 - a. Mempunyai nama yang sama dengan class
 - b. Tidak mempunyai return type

```
public class Siswa{  
    private int nrp;  
    private String nama;  
    public Siswa(int n, String m){  
        nrp = n;  
        nama = m;  
    }  
}
```

3. Apakah yang dimaksud dengan overloading constructor?
Overloading constructor adalah kejadian pada suatu class yang mempunyai lebih dari satu konstruktor, dengan tiap konstruktor memiliki parameter yang berbeda, bisa berbeda jumlah parameternya ataupun berbeda type data parameternya

B. LATIHAN

1. Mengimplementasikan UML class diagram dalam program untuk class kalender



- Listing Program Class Kalender

```
public class Kalender{  
    private int tanggal = 1;  
    private int bulan = 1;  
    private int tahun = 2000;  
  
    public Kalender(int tanggal){  
        this.tanggal = tanggal;  
    }  
}
```

```

public Kalender(int bulan, int tahun){
    this.bulan = bulan;
    this.tahun = tahun;
}

public Kalender(int tanggal, int bulan, int tahun){
    this.tanggal = tanggal;
    this.bulan = bulan;
    this.tahun = tahun;
}

public void setTanggal(int tanggal){
    this.tanggal = tanggal;
}

public int getTanggal(){
    return tanggal;
}

public void setBulan(int bulan){
    this.bulan = bulan;
}

public int getBulan(){
    return bulan;
}

public void setTahun(int tahun){
    this.tahun = tahun;
}

public int getTahun(){
    return tahun;
}
}

```

- Listing Program TesKalender

```

public class TesKalender {
    public static String getTime(Kalender kal) {
        String tmp;
        tmp = kal.getTanggal() + "-" + kal.getBulan() + "-" + kal.getTahun();
        return tmp;
    }
    public static void main(String args[]) {
        Kalender kal = new Kalender(8);
        System.out.println("Waktu awal : " + getTime(kal));
        kal.setTanggal(9);
        System.out.println("1 hari setelah waktu awal : "+getTime(kal));
        kal = new Kalender(6, 2003);
        System.out.println("Waktu berubah : " + getTime(kal));
        kal.setBulan(7);
        System.out.println("1 bulan setelah itu : " + getTime(kal));
        kal = new Kalender(20, 10, 2004);
    }
}

```

```

        System.out.println("Waktu berubah : " + getTime(kal));
        kal.setTahun(2005);
        System.out.println("1 tahun setelah itu : " + getTime(kal));
    }
}

```

Hasil Compile

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 8>javac TesKalender.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 8>java TesKalender.java
Waktu awal : 8-1-2000
1 hari setelah waktu awal : 9-1-2000
Waktu berubah : 1-6-2003
1 bulan setelah itu : 1-7-2003
Waktu berubah : 20-10-2004
1 tahun setelah itu : 20-10-2005

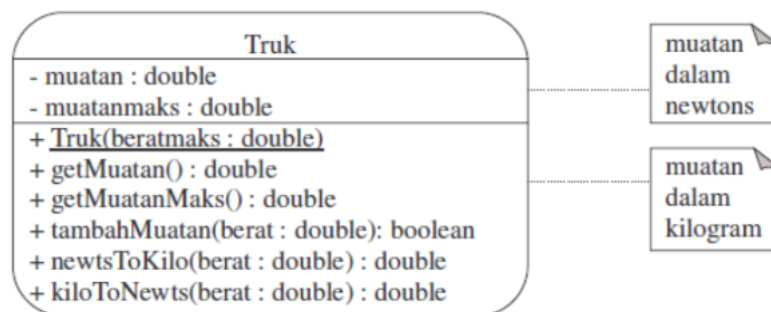
C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 8>_

```

Analisa :

Dalam pengimplementasian enkapsulasi terhadap class diagram kalender, atribut yang terdapat pada class diagram tersebut terlebih dahulu diubah modifiernya menjadi private, dan dilakukan metode setter and getter untuk melakukan pengaksesan terhadap atribut – atribut class diagram tersebut. Kemudian dapat dilihat pada program di atas, terjadi overloading constructor, dikarenakan terdapat lebih dari satu konstruktor, namun konstruktor – konstruktor tersebut memiliki jumlah parameter yang berbeda

2. Mengimplementasikan UML class diagram dalam program untuk class truk



- Listing Program Class Truk

```

public class Truk{
    private double muatan;
    private double muatanmaks;

    public Truk(double beratmaks){
        muatanmaks = beratmaks;
    }

    public double getMuatan(){
        return newtsToKilo(muatan);
    }
}

```

```

public double getMuatanMaks(){
    return muatanmaks;
}

public boolean tambahMuatan(double berat){
    if((newtsToKilo(muatan) + berat) < muatanmaks){
        muatan += kiloToNewts(berat);
        return true;
    }
    else{
        return false;
    }
}

public double newtsToKilo(double berat){
    berat = berat /9.8;
    return berat;
}

public double kiloToNewts(double berat){
    berat = berat * 9.8;
    return berat;
}
}

```

- Listng Program TesTugas2

```

public class TesTugas2{
    public static void main(String args[]){
        boolean status;

        Truk truk = new Truk(900);
        System.out.println("Muatan maksimal = "+truk.getMuatanMaks());
        status = truk.tambahMuatan(500.0);
        System.out.println("Tambah muatan : 500");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");

        status = truk.tambahMuatan(300.0);
        System.out.println("Tambah muatan : 300");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");

        status = truk.tambahMuatan(150.0);
        System.out.println("Tambah muatan : 150");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
    }
}

```

```

        status = truk.tambahMuatan(50.0);
        System.out.println("Tambah muatan : 50");
        if (status)
            System.out.println("Ok");
        else
            System.out.println("Gagal");
        System.out.println("Muatan sekarang = " + truk.getMuatan());
    }
}

```

Hasil Compile

```

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 8>javac TesTugas2.java

C:\Users\ratri\OneDrive\Documents\OOP\Praktikum 8>java TesTugas2.java
Muatan maksimal = 900.0
Tambah muatan : 500
Ok
Tambah muatan : 300
Ok
Tambah muatan : 150
Gagal
Tambah muatan : 50
Ok
Muatan sekarang = 849.9999999999999

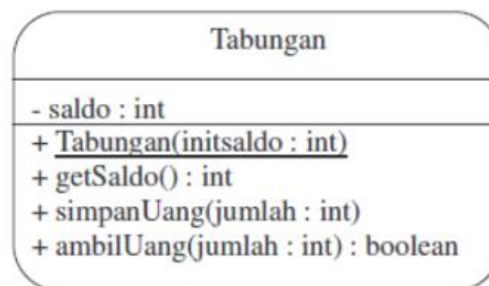
```

Analisa :

Program di atas digunakan untuk melakukan pengimputan terkait muatan suatu truk

C. TUGAS

1. Menerapkan konsep enkapsulasi pada kelas tabungan yang terdapat di tugas 1



- Listing Program Class MultiTabungan

```

public class MultiTabungan{
    private int saldo;

    public MultiTabungan(int initsaldo){
        saldo = initsaldo;
    }

    public int getSaldo(){
        return saldo;
    }

    public void simpanUang(int menu, int jumlah){

```

```

        if(menu == 1){
            int total = jumlah * 10000;
            saldo += total;
        }
        else if(menu == 2){
            int total = jumlah * 9000;
            saldo += total;
        }
        else{
            saldo += jumlah;
        }
    }

    public boolean ambilUang(int menu, int jumlah){
        if(saldo > jumlah){
            saldo -= jumlah;
            return true;
        } else {
            return false;
        }
    }

    public int konvUang(int menu, int uang){
        if(menu == 1){
            return uang * 9000;
        }
        else if(menu == 2){
            return uang * 10000;
        }
        else{
            return uang;
        }
    }
}

```

- Listing Program TesMulti Tabungan

```

import java.util.Scanner;

public class TesMultiTabungan{
    public static void main(String args[]){
        boolean status;
        MultiTabungan tabungan = new MultiTabungan(500000);
        System.out.println("Saldo : " + tabungan.getSaldo());

        System.out.println("1. US");
        System.out.println("2. AUD");
        System.out.println("3. IDR");

        Scanner sc = new Scanner(System.in);
        System.out.print("Pilih menu : ");
        int menu = sc.nextInt();
        System.out.print("Simpan uang : ");
        int simpan = sc.nextInt();
    }
}

```

```
        tabungan.simpanUang(simpan, menu);
        System.out.println("Jumlah uang disimpan : " +
tabungan.konvUang(simpan, menu));

        System.out.print("Ambil uang : ");
        int ambil = sc.nextInt();
        status = tabungan.ambilUang(ambil);
        System.out.println("Jumlah uang diambil : " + ambil);

        if(status){
            System.out.println("Ok");
        } else {
            System.out.println("Gagal");
        }
        System.out.println("Saldo : " + tabungan.getSaldo());
    }
}
```